

Talentua: entorn universitari en el marc del campus virtual i del *matching* empresa-estudiant (part I)

Alan Fusté Rodríguez

Resum– Talentua és un projecte que combina dues plataformes: Smart Campus, una plataforma LCMS per a centres d'estudis, i Talentua, una plataforma de *matching* entre estudiants i empreses. El projecte s'ha dividit en sis TFG, enfocats a diferents aspectes i s'aborda des d'una perspectiva d'emprenedoria amb l'objectiu que esdevingui una *startup* en un futur.

Aquest treball se centra en Smart Campus i té com a objectiu que s'arribi a obtenir un mínim producte viable que es pugui provar en un entorn de producció i assenti una base sòlida del LCMS del futur. S'han desenvolupat tant el *backend* com el *frontend* utilitzant tecnologies molt punteres com Django, React i GraphQL. S'han integrat dades reals de milers d'estudiants per tal de provar la plataforma i demostrar la viabilitat tècnica de la mateixa quan es treballa amb un volum de dades d'un entorn de producció.

Paraules clau– Campus virtual, LCMS, Django, React, Dades, GraphQL, Estudiants, Universitats, Escoles, Centres d'estudis, Talent, Backend, Frontend, Disseny, Futur.

Abstract– Talentua is a project that combines two platforms: Smart Campus, a platform LCMS for study centres, and Talentua, a platform of matching among students and companies. The project has divided at six TFG, focused at different aspects and accosts since a perspective of entrepreneurship with the aim that becomes a startup in the future.

This paper focuses at Smart Campus and has as an aim that reaches obtain a minimal viable product that can be proved at some surroundings of production and settle a solid base of the LCMS of the future. There have been developed the backend and the frontend using modern technologies like Django, React and GraphQL. There have been integrated real data of thousands of students for such to prove the platform and evidence the technical feasibility of it when works with a volume of data of some surroundings of production.

Keywords– Virtual campus, LCMS, Django, React, Data, GraphQL, Students, Universities, Schools, Study centres, Talent, Backend, Frontend, Design, Future.

1 INTRODUCCIÓ

Al llarg dels anys a la universitat s'han anat detectant problemes i carències en ambdues plataformes de campus virtual que s'han utilitzat: una pròpia de la universitat i una basada en Moodle. Tot i les aparents millores, les plataformes estan lluny de facilitar l'ús a estudiants i docents ja que no incorporen eines ni

mecanismes que agilitzin completar processos. Manquen funcionalitats com guardat automàtic, consells d'acció o inclús accions automàtiques com, per exemple, enviar les notes automàticament quan s'acaba el termini d'una entrega, etcètera.

A més a més, s'ha detectat que la universitat, la UAB, tot i els seus plans estratègics com l'Horitzó 2030, no està sent capaç d'adaptar-se prou ràpid a les necessitats de la societat. L'ús de tècniques de *big data* i d'intel·ligència artificial, donarien la capacitat a la universitat a resistir la creixent influència i guany d'estudiants dels nous models de formació *online*, ja que compta amb una posició que encara és privilegiada i un volum d'estudiants molt superior a altres models de formació.

- E-mail de contacte: alan.fuste@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Rubén Rubio Barrera (dEIC)
- Curs 2019/20

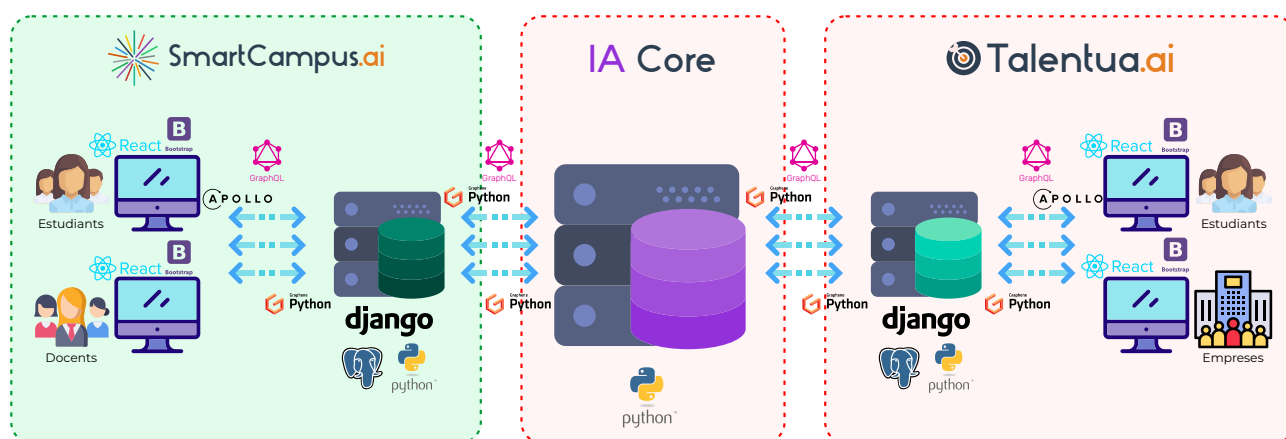


Fig. 1: Arquitectura web de les plataformes, en verd la part en la que es centra aquest treball.

Al juny de 2018, es decideix arrancar el projecte Talentua conjuntament amb en Xavier Velasco i en Dyhagho Briceño, ambdós companys de la doble titulació i que realitzen les parts II i III, respectivament. Com es pot veure a la taula 1, el conjunt global del projecte en el marc del TFG no queda aquí, ja que durant el segon semestre es realitzaran tres treballs més, corresponents al Grau en Sistemes de Telecomunicació, que es focalitzaran en l'ús d'intel·ligència artificial que permeti situar la plataforma a l'avantguarda tecnològica: usant tecnologies i tècniques modernes i incorporant una capa de processament de dades i d'intel·ligència artificial.

| Part | Autor | Grau | Àmbit |
|------|---------|------|---------------------------|
| I | Alan | GEI | Web |
| II | Xavier | GEI | Web |
| III | Dyhagho | GEI | DevOps + CI |
| IV | Alan | GEST | Intel·ligència Artificial |
| V | Xavier | GEST | Intel·ligència Artificial |
| VI | Dyhagho | GEST | Intel·ligència Artificial |

TAULA 1: LLISTA DELS SIS TFG DE TALENTUA

Aquests sis treballs són proposats pels tres estudiants i, per tant, són d'iniciativa pròpia. Donat que el temps de desenvolupament d'una solució com la que es detalla a continuació és gran, es va optar per fer aquesta proposta de treballs. A més a més, inclouen una part de treball en equip, que requereix de coordinació per totes les parts, i s'inclou una visió des del punt de vista de l'emprenedoria, fent que no només es tracti d'una solució viable tècnicament.

L'objectiu principal d'aquests sis treballs és aconseguir un producte mínim viable (MVP). Aquesta primera versió, que s'estima estigui llesta durant l'estiu de 2020, no és una versió comercialitzable però sí serà una versió que es pugui fer servir en un entorn controlat, a mode de *beta*, i que pugui provar-se amb algun sector implicat. Per tant, els objectius que a continuació es detallen i s'analitzen s'han d'entendre des de la perspectiva d'un projecte no només en el marc de sis treballs de final de grau, sinó sota un full de ruta que busca que Talentua esdevingui una societat plenament operativa.

La solució presentada es compon de dues plataformes interconnectades: una destinada a funcionar com a campus

virtual (Smart Campus) i una altra amb l'objectiu de realitzar un *matching* acurat entre estudiants i empreses per tal d'aconseguir que els estudiants trobin el lloc de treball en el que millor encaixen (Talentua).

En la figura 1 es veu la integració de Smart Campus i Talentua amb el mòdul d'intel·ligència artificial que processa les dades que es van generant des de Smart Campus. Un factor important, que en principi no suposarà un bloc per si sol de cap part però sí esdevindrà un element transversal i bàsic de les plataformes, és la privacitat. L'objectiu d'obtenir dades és proporcionar un valor afegit a qui les genera, en aquest cas l'estudiant. Per tant, l'objectiu no és en cap cas vendre les dades o donar-ne accés a tercers. Aquest és un punt fonamental i serà capital durant el desenvolupament de les plataformes.

En el present document s'introdueix al lector en el projecte de manera que pugui entendre les motivacions del mateix. Aquesta primera part se centra en el desenvolupament del Smart Campus i, per tant, Talentua i la intel·ligència artificial queden pendents per un futur.

A la secció 2, es poden trobar els objectius del treball.

Abans d'entrar en el nucli del treball, a la secció 3, es pot trobar l'estat de l'art.

Respecte al desenvolupament del treball, s'hi poden trobar la metodologia emprada (secció 4) i els resultats del treball (secció 5). Aquests apartats ajudaran al lector a entendre com s'ha treballat els darrers mesos a nivell d'equip i a veure quins són els fruits d'aquest treball.

Tot seguit es poden trobar les conclusions (secció 6) d'aquesta part, així com la bibliografia consultada per a realitzar la present memòria.

Per últim, es troba la secció 7, en la que s'analitza el futur de Talentua des d'una perspectiva no només tecnològica, sinó des d'un punt de vista emprenedor seguint el model Canvas.

1.1 Smart Campus

Smart Campus és la plataforma LCMS que es proposa i que es caracteritza per ser modular, intel·ligent i pensada per facilitar al màxim la feina de docents i estudiants. Compta amb un gran ventall de funcionalitats que es detallen a les seccions 2.1 i 5.

Utilitzant un model de comercialització *freemium* amb diferents paquets de funcionalitats i nombre d'estudiants,

es podria replicar la plataforma Smart Campus adaptant-la a diferents centres educatius i assegurant que el seu cost sigui considerablement menor al cost actual (en referència al cost d'utilitzar Moodle a la UAB tal com es detalla a la secció 3.1.1).

1.2 Talentua

Talentua és la plataforma de *matching* entre estudiants i empreses, amb ella es busca que els estudiants treballin en empreses on encaixin el màxim possible i, recíprocament, que les empreses tinguin accés a estudiants que compleixin el perfil que busquen.

A la secció 3.2 s'introdueix un concepte clau: el talent. Es considera que el talent no és únic i s'afirma, amb prou seguretat, que hi ha persones amb unes grans capacitats per certs aspectes d'un àrea de coneixement i/o d'altres amb unes aptituds determinades. Aquestes capacitats inclús poden veure's reflectides en forma de nota si s'analitza el detall, per exemple: la nota d'un examen, la nota d'una presentació oral, la nota d'una assignatura, etcètera. Hi ha un factor més que és la normalització de les notes: cal posar-les en context amb la resta de l'alumnat, amb la facultat i universitat on es cursen, etcètera. Tots aquests elements s'analitzaran a fons en un futur en les parts IV, V i VI del projecte. Les universitats també podrien treure profit d'aquesta eina ja que podrien col·locar millor, laboralment, als seus estudiants, així com obtenir informació sobre l'evolució dels mateixos i les necessitats en temps real del món laboral.

2 OBJECTIUS

Les parts I i II del projecte s'han centrat en l'entorn web del Smart Campus i s'han desenvolupat les funcions que marquen una diferència més gran amb les funcionalitats LCMS existents. Per tant, l'objectiu principal d'aquest treball és aconseguir un MVP de Smart Campus. Les funcionalitats que es fixen com a bàsiques i, per tant, que s'engloben dins del MVP són les llistades a la secció 2.1.

2.1 Objectius inicials

Les funcionalitats que es van definir inicialment per a realitzar en les parts I i II són:

- Notes
- Calendari (classes, entregues i exàmens; que sigui personalitzat i ampliable individualment)
- Repositori de material dels estudiants
- Esdeveniments
- Sincronització amb Talentua
- Informes de les assignatures que incloguin aspectes a millorar, estudiants destacats, estudi de les notes, etcètera.
- Sistema d'enquestes
- Mòdul de denúncia anònima (enfocat a l'assetjament)

D'altres funcionalitats com les Assignatures, Professors, Estudiants, etcètera s'han desenvolupat parcialment però, per exemple, no s'ha inclòs un entorn d'administrador propi (més enllà del de Django, veure secció 3.3) que permeti gestionar específicament aquests perfils per part de la direcció del centre.

Les tasques corresponents a aquestes funcionalitats i als models de dades es va detallar al ClickUp (veure secció 4) del projecte (parts I i II), es va estimar el seu temps, *sub-tasques* i es van configurar les dependències entre tasques. Cal notar que inicialment es van planificar tasques en base a totes les funcionalitats que es requerien per tenir un LCMS complet i superior a les plataformes existents (veure secció 3). Les hores requerides per a implementar aquestes funcionalitats amb una bona qualitat no feia viable encabir-les totes en les dues primeres parts, per això es van triar les funcionalitats requerides per a fer un MVP. Aquest procés va suposar que un 45% de les tasques que s'havien planificat es descartessin.

La part III del treball s'ha centrat en l'entorn de producció de les plataformes, que usen les tecnologies més modernes i estan allotjades al servei *cloud* Amazon Web Services. A més a més, s'ha treballat des d'una perspectiva de *DevOps*¹ i el *CI*² fent que, per exemple, quan es puja nou contingut a una determinada branca del repositori de codi aquest automàticament es publiqui al *cloud* si passa una sèrie de tests.

3 ESTAT DE L'ART

3.1 Programari de campus virtual

Principalment hi ha dos tipus de programaris de campus virtual: els LMS i els LCMS. La diferència entre ambdues és molt subtil i de vegades s'engloben totes dins de les sigles LMS (tot i ser LCMS).

Un LMS (*Learning Management System*) és un programari que facilita l'educació en un entorn en línia gràcies a que proveeix una plataforma connectada des de la que es poden gestionar totes les dades i processos necessaris. Un LCMS (*Learning Content Management System*) és un LMS però que a més integra un CMS (*Content Management System*) i, per tant, permet organitzar i crear tot el contingut necessari per a realitzar aquesta educació en línia.

A la figura 2 es poden veure les principals plataformes de LCMS en base al seu ús als Estats Units. El mercat d'aquest tipus de programari s'estima que serà de 15.720 milions de dòlars l'any 2021 [3]. L'Amèrica del Nord sembla que s'endurà la major part d'aquest benefici [4] ja que dos dels principals programaris (Blackboard i Canvas) són propietat d'empreses nord-americanes. El mercat dels LCMS no és només exclusiu d'entorns escolars o universitaris ja que el 41.7% de les empreses de *Fortune 500* a nivell global utilitzen algun programari de LCMS per formar als seus empleats [5].

¹El desenvolupament d'operacions (*DevOps*) és una metodologia de desenvolupament del programari que posa èmfasi en la comunicació, la col·laboració, la integració, l'automatització i la mesura del nivell de cooperació entre desenvolupadors.

²La integració contínua (*CI*) és una pràctica de desenvolupament de programari en la qual els membres de l'equip integren el seu codi amb freqüència i, per cada integració, es realitza una verificació automàtica per detectar errors.

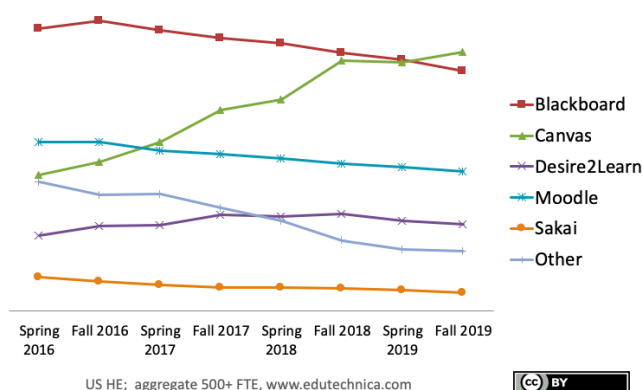


Fig. 2: Evolució de l'ús de plataformes LCMS als Estats Units

3.1.1 Moodle

Una de les principals plataformes LCMS de codi obert en l'actualitat és Moodle, programada en PHP i nascuda a Austràlia l'any 2001. Aquesta és la base que sustenta en l'actualitat el Campus Virtual de la UAB i de molts centres d'estudis catalans. Hi ha dos grans avantatges de Moodle respecte a la seva competència i és que és de codi obert i que té una gran comunitat al darrera.

Moodle s'ha estancat tecnològicament: no és una eina fàcil d'utilitzar, no disposa de les últimes tecnologies ni està a l'avantguarda del que la comunitat educativa necessita. Tampoc és una eina que es pugui qualificar com a intel·ligent, ja que no dona facilitats de cap tipus als usuaris ni utilitza tècniques modernes d'intel·ligència artificial per millorar aspectes del dia a dia d'estudiants, docents o personal administratiu.

Tot i ser una eina gratuïta, a la pràctica no ho és: la UAB pressuposta en el campus virtual per graus, per postgraus i inversions en el mateix, 1.221.606 euros anuals [1]. El desglossament d'aquest pressupost es pot veure a la taula 2 i el concepte és per tots els casos "Àrea de Planificació de Sistemes d'Informació (Campus Virtual)". Aquesta inversió és molt considerable partint de la base que la llicència de Moodle és gratuïta i demostra que hi ha un gran marge per reduir aquesta partida.

| Programa | Import |
|---|--------------------|
| Programa de docència de grau | 769.612 € |
| Postgrau i formació contínua | 207.673 € |
| Gestió d'inversions en tecnologia i comunicacions | 244.321 € |
| Total | 1.221.606 € |

TAULA 2: PARTIDES DEL PRESSUPOST DE 2019 DE LA UAB QUE FAN REFERÈNCIA EXPLÍCITA AL CAMPUS VIRTUAL

3.1.2 Blackboard

Blackboard LMS és una plataforma d'educació a distància que tradicionalment s'ha utilitzat de manera majoritària a les escoles i universitats dels Estats Units. Té més de 20 anys però, tot i que es caracteritza per ser molt robusta, ha

tingut diversos problemes de seguretat greus i incidents vers la privacitat dels estudiants.

Aquest programari no és modern i, tot i que va rebent actualitzacions, els canvis són lents per tal d'assegurar que sigui robust. Un tret fonamental de Blackboard és que és privatiu, per tant, cal pagar la llicència que autoritza el seu ús i el seu preu és d'uns 1000 euros per usuari [2].

Per tant, si es volgués implementar Blackboard a la UAB, una universitat pública de més de 37.000 estudiants i 3.000 professors, el cost seria d'uns 40 milions d'euros (entre un 10 i un 15% del pressupost anual de la universitat).

3.1.3 Canvas

Aquest programari LMS va ser creat per l'empresa de tecnologies del món educatiu Instructure Inc. l'any 2011 per dos estudiants recent graduats i que actualment es fa servir en més de 3000 universitats, districtes escolars i institucions de tot el món.

Canvas està programat en Ruby on Rails com a *framework* de *backend* amb una base de dades PostgreSQL. A nivell de *frontend* utilitza JQuery, HTML5 i CSS3. També es caracteritza per l'ús d'*OAuth* per tal de proveir una millor integració amb xarxes socials com Facebook o Twitter.

Es pot instal·lar amb diferents proveïdors de *cloud* i, a més a més, es comercialitza seguint un model *SaaS* (programari com a servei o *Software as a Service*) allotjat a Amazon Web Services.

3.2 El talent

Es considera que en l'actualitat no existeix un problema de falta de talent, sinó que el dèficit es troba en l'escassetat d'iniciatives per potenciar-lo. Actualment, les úniques promocions que es fan dels talents són en base a beques que es destinen als estudiants amb millor currículum, és a dir, millor nota mitjana. En una època on s'ha normalitzat parlar d'intel·ligències múltiples, la definició de talent probablement hauria de ser més àmplia.

Les solucions que s'usen per trobar feina són majoritàriament digitals, llevat de les tradicionals agències privades o públiques, però totes aquestes solucions tenen dos problemes comuns: no verifiquen les dades que es proporcionen per part del possible empleat i aquestes són genèriques. Per exemple: Graduada en Enginyeria Informàtica per la UAB, no dona cap detall, com a molt, un cop més, la nota mitjana. Tenen totes les persones graduades en Enginyeria Informàtica les mateixes capacitats? I aptituds? Es això una bona mesura del talent?

Per una banda, les universitats no tenen cap manera, més enllà d'enquestes de l'Alumni, de saber com van els seus estudiants ni tampoc manera de saber quins perfils exactes requereix el món laboral, més enllà d'algunes empreses determinades que tenen relacions estretes amb les universitats. Per altra banda, les empreses no tenen manera de comprovar que els currículums són verídics i que no han estat alterats, a més a més, han de decidir-se per una persona després d'un costós procés de reclutament i selecció que no té garanties d'èxit (sovint són processos recurrents o, fins i tot, continus en el temps) ja que no es coneix a fons el perfil del potencial contractat.

3.3 Tecnologies

La tria del *stack* tecnològic d'un programa és un element essencial que no només condiciona el procés de desenvolupament sinó que té un impacte en la solució final. El *stack* de Talentua havia de ser modern, amb tecnologies punteres però alhora havia de ser fàcil d'implementar donat que, com el desenvolupament s'encabia en el TFG, la corba d'aprenentatge havia de ser assumible. Hi ha dues variables més que s'han tingut en compte a l'hora de triar les tecnologies i és que, al ser un projecte que arribarà a un entorn real, aquestes han de comptar amb un número considerable de programadors per tal de fer viable pagar els sous i, per últim, han de ser tecnologies fàcils de mantenir, amb anys de suport per davant i correccions freqüents (s'ha estudiat l'activitat a GitHub del codi oficial de cada tecnologia).

S'han estudiat diferents llenguatges de programació (PHP, JavaScript, Java, Python, Ruby i Scala) pel *backend* i el *frontend*, *frameworks* de CSS per a l'apartat visual, tecnologies de bases de dades SQL i tecnologies de connexió entre plataformes (API). Tota la informació està disponible a l'apèndix A.1 on es pot llegir un informe amb una anàlisi de més de trenta tecnologies.

Un cop realitzat l'anàlisi s'ha optat per utilitzar Django 3, un *framework* de Python com a *backend*, ja que al tenir experiència prèvia amb ell s'accelera la implementació d'aquest. A més a més, és una tecnologia moderna, ràpida i robusta.

Tot i que Django té un sistema de *templates* de HTML5 molt àgil basat en Jinja2 s'ha optat per una solució de *frontend* basada en JavaScript, ja que són més amigables amb l'usuari, permeten un nivell visual (UI) i d'experiència d'usuari (UX) més avançat, perquè solen incloure eines que faciliten la creació d'aplicacions mòbils, etcètera.

Com a base de dades s'utilitza una especificació de SQL, Postgresql que s'ha triat per la seva robustesa i escalabilitat treballant amb bases de dades de gran mida enfront a altres tipus de SQL.

Angular és una tecnologia més completa que React sobre el paper i així s'exposa a l'apèndix A.1 però el defecte que té és que és complex: l'ús de *Typescript* complica la seva implementació. Per tant, s'ha optat per React, un *framework* desenvolupat per Facebook molt contrastat i sòlid en la seva versió 16.

Com a complement a React, al *frontend*, s'ha utilitzat Bootstrap 4 ja que va més enllà d'un *framework* CSS en incloure moltes funcionalitats de JavaScript. A més a més, es complementa molt bé amb React permetent treballar amb Sass i no directament amb CSS.

Una de les dificultats d'emprar dues tecnologies diferents en el *frontend* i en el *backend* és que cal connectar-los d'alguna manera. Una de les grans avantatges de Django és que incorpora un sistema, anomenat Django Rest Framework, que permet desplegar una API dels models de manera fàcil.

Les APIs de tipus REST permeten crear un punt d'accés al *backend*, un *endpoint*. El problema que tenen és que són costoses de mantenir i no són eficients. Si, per exemple, es mostra en el *frontend* el perfil d'un usuari, la implementació més comuna és generar un *endpoint* amb l'API REST i que el *backend* envii al *frontend* tot l'objecte usuari; el problema és que el *frontend* potser només vol el nom de l'usuari i no tota la resta d'informació que sí o sí s'està enviant, això

suposa un consum de dades, d'energia i de temps innecessari.

Hi ha un altre tipus d'API anomenada GraphQL que permet fer crides dinàmiques des del *frontend* (utilitzant la llibreria Apollo) fins al *backend* (utilitzant la llibreria GraphQL), de manera que en cada moment es demana la informació que s'ha d'utilitzar. A més a més, permet mantenir variables actualitzades i d'altres característiques que fan que sigui molt útil per connectar Django i React.

Com es pot veure a la figura 1, hi ha moltes connexions entre les diferents plataformes i entre els *frontends* i els *backends*, amb una sola implementació de GraphQL en cada *backend* es poden realitzar totes les connexions de manera dinàmica i segura.

4 METODOLOGIA

En començar el treball es va fixar que durant el desenvolupament se seguiria una metodologia àgil, concretament SCRUM, tot i que amb certes peculiaritats per adaptar-lo al marc del treball de final de grau. Es va estipular que es realitzarien esprints de dues setmanes que anirien lligats a les reunions que es produirien cada dues setmanes amb els tutors (qui han fet de *Product Owner*).

A més a més, es va concretar que es realitzarien reunions diàries entre els tres membres de l'equip per tal de veure com avança el projecte (a través de WhatsApp o similars) i que s'utilitzaria Microsoft Teams per a organitzar-se i tenir un espai comú amb els tutors, així com l'ús de ClickUp per a la gestió de les tasques i els esprints. A la figura 3 es pot veure un exemple de l'ús del ClickUp.

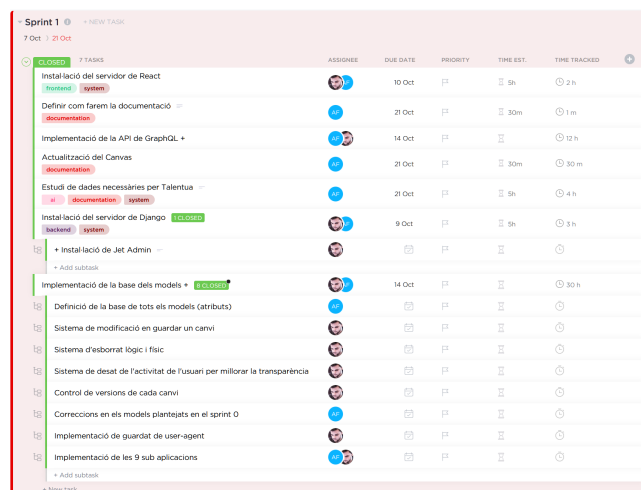


Fig. 3: Tasques del primer esprint al ClickUp un cop havia acabat

Degut a que s'han de realitzar treballs individuals no es pot fer una selecció de tasques tal com sol marcar SCRUM però sí s'ha consensuat els temps de duració de les tasques. Per tal de gestionar els esprints de SCRUM s'ha utilitzat la plataforma ClickUp, amb el pertinent *backlog*. Per últim, quan un mòdul dels establerts ha estat llest, en Dyhaghó Briceño (que és qui fa la tercera part del projecte) l'ha pujat a l'entorn de producció.

S'ha seguit la metodologia prevista i s'ha treballat de manera fluida i còmoda. L'ús de ClickUp com a organitzador de tasques ha funcionat molt bé i ha facilitat molt la feina.

La duració de l'esprint en dues setmanes es considera encertat, ja que ha aportat molt el fet de realitzar una reunió comuna dels tres treballs de final de grau.

A més a més, s'ha dissenyat i implementat seguint el paradigma de *Domain Driven Design* (DDD) que, tot i ser més complex, ha permès portar el disseny de software a un altre nivell creant un programari més escalable i fàcil de mantenir. Això és possible ja que DDD busca millorar la comunicació entre desenvolupadors i experts en el domini (que és un dels principals problemes que s'afronten en un projecte com aquest) fent que el desenvolupament estigui guiat pel domini i es busca sempre que es respecti aquest al màxim. Per això que se segueix una arquitectura de capes basada en el domini i, alhora, en base a desacoblar les diferents capes s'aconsegueix millorar l'escalabilitat i el manteniment del codi. Es pot veure l'estructura de DDD en la figura 4.

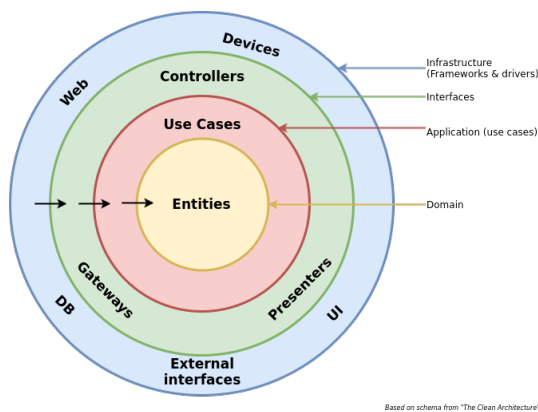


Fig. 4: Esquema de DDD

5 RESULTATS

S'han fet avenços en molts aspectes, principalment en la definició de la base del *backend* i del *frontend*, així com una part de tractament de dades reals d'estudiants i l'exposició de les dades.

5.1 Backend

Smart Campus compta amb un *backend* funcional amb característiques que en molts aspectes superen a les de Moodle i que poden afavorir polítiques de transparència i augmentar la credibilitat del centre educatiu que opti per utilitzar-lo, gràcies a l'ús d'un sistema de control de versions en tots els elements.

5.1.1 Control de canvis

Tots els models de Django que s'han definit estan pensats per ser fàcils de mantenir, consistents, que es puguin traduir a diferents llengües, etcètera. Smart Campus va molt més enllà de tot això ja que s'ha implementat un sistema que permet que quan es realitza un canvi en qualsevol element es guarda una còpia de l'antiga versió. Per tant, s'ha aconseguit tenir un sistema de control de versions de tot el que passa a la plataforma: si es canvia una nota es pot saber quina nota hi havia abans, qui l'ha canviada, des d'on s'ha canviat, etcètera. Això és rellevant per aspectes com les

notes, però també permetrà oferir extres com que els professors puguin recuperar entregues prèvies dels alumnes en cas d'incidències (si s'esborra alguna cosa per error es pot recuperar). Aquest sistema també guarda qui, quan i des d'on (es desa el *user-agent*³) s'han realitzat les modificacions, això permetrà un nivell extra de seguretat als centres i facilitarà la detecció i prova d'intents de frau en notes o entregues.

A la figura 5 es pot apreciar el llistat de canvis que s'han realitzat sobre una nota. A la figura 6 es pot observar el detall d'una nota que s'ha modificat i com queden registrats els canvis que s'han realitzat en el panell d'administració de Smart Campus que s'introdueix a la secció 5.1.4.

| FECHA/HORA | USUARIO | ACCIÓN |
|----------------------------------|-----------|--|
| 8 de Febrero de 2020 a las 12:22 | alanfuste | Añadido. |
| 8 de Febrero de 2020 a las 12:23 | alanfuste | Modificado verbose_name, text_value y numeric_value. |

Fig. 5: Històric de versions d'una nota

| DATA | ACTIVE OBJECT | INFORMATION | TIMESTAMP |
|---|----------------------|-----------------------------|-----------|
| Data: | | | |
| <pre>{'name': 'aprovat', 'type': None, 'lesson': None, 'student': '247ab239-b329-4d32-a523-8864e075de96', 'activity': None, 'created_by': None, 'is_deleted': False, 'text_value': 'C', 'user_agent': None, 'modified_by': None, 'verbose_name': 'Aprovat', 'numeric_value': 5.0, 'subject_period': 'b4ff5c76-23ab-49d4-8e1a-a94f84329016'}</pre> | | | |
| GRABAR | Grabar y añadir otro | Grabar y continuar editando | ELIMINAR |

Fig. 6: Detall d'una versió antiga d'una nota

Smart Campus té un sistema d'esborrat poc convencional, ja que quan s'esborra un element del web l'usuari no el veu més però aquest continua guardat a la base de dades. Aquest fet, que s'anomena esborrat lògic, aporta avantatges en seguretat i pot ser útil per a mediacions en casos de conflictes. Com a complement d'aquesta funcionalitat s'ha afegit un mode per realitzar un esborrat físic i que, per tant, desapareguin els registres de la base de dades (aquest mode està només disponible pels administradors).

5.1.2 Identificadors

Com hi ha tants elements guardats a la base de dades s'ha utilitzat un identificador únic universal UUID (*Universally unique identifier*) que és un número de 16 bytes (el número de possibles UUID és de 16^{32}) que permet que s'identifiqui cada registre de la base de dades.

5.1.3 Característiques

A Smart Campus es poden gestionar des d'institucions fins a les entregues dels estudiants i les seves notes. És una apli-

³El *user-agent* conté la informació de qui realitza una connexió web: IP, navegador, etcètera.

cació que s'ha pensat per a que es puguin gestionar diferents tipus d'institucions o centres educatius i que sigui escalable. S'ha pensat per a que sota una sola plataforma puguin con- viure diferents institucions permetent que, per exemple, hi hagués un Smart Campus per a totes les escoles d'un poble o per a un grup d'universitats, però també podria utilitzar- se per una sola institució i que cada centre d'estudis tingués una instal·lació. Aquestes dues maneres de funcionar per- meten que Smart Campus es pugui comercialitzar de diver- ses maneres i, per tant, tinguí més opcions de ser viable.

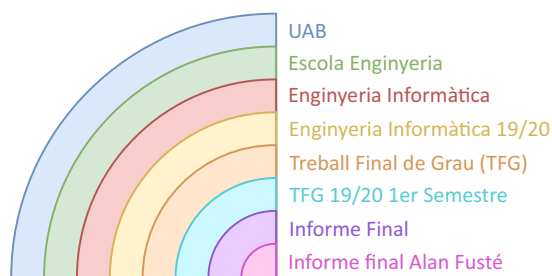


Fig. 7: Exemple de l'estructura de Smart Campus si s'en- tregués aquest informe

Entrant en més detall, tal com es veu a la figura 7 i pre- nent d'exemple un hipotètic Smart Campus de la UAB, es declaren com a centres cadascuna de les facultats de la uni- versitat. A cadascuna se li associen els graus, cursos i màsters que s'hi imparteixen. En cada curs acadèmic es crea una nova instància del grau que es especifica d'aquell curs. Aquesta manera d'organitzar la informació dels graus facilita l'anàlisi de dades i comparativa entre centres, per- met categoritzar molt més la informació i alhora permet se- guir un ordre molt intuïtiu.

Cada grau té associades una sèrie d'assignatures que, de nou, tenen una instància per cada any en què es realitzen. Seguint l'exemple de la figura 7 es té una assignatura Tre- ball de Final de Grau amb una instància pel TFG del present curs i pel primer semestre. A dins de cada instància d'as- signatura es poden afegir molts elements, com bibliografia, publicacions, fitxers, etcètera i, a més a més, s'hi poden de- finir esdeveniments, lliçons i activitat.

En l'esmentat exemple (figura 7) es defineix una activitat pel present Informe Final. A Smart Campus quan es crea una activitat es creen instàncies d'aquesta per a cada estu- diant o grup d'estudiants, segons sigui el cas, en les quals es poden associar entregues. En aquest següent nivell, a l'exemple, es realitza l'entrega del present informe final.

Un cop realitzada una entrega, dins del període establert, el professor associat pot entrar i introduir una nota a l'en- trega que, automàticament i segons les ponderacions indi- cades, s'aplica a la nota final de l'assignatura. En el cas que es realitzi un treball en grup, es pot definir la mateixa nota per tots els estudiants o també definir una nota específica per a cada estudiant. Això agilitza el procés d'entrada de notes, però també deixa marge al docent per posar notes in- dividuais quan sigui necessari. Les notes es poden introduir en format de text o numèric i automàticament es conver- teixen al format que no s'ha introduït, això fa que Smart Campus es pugui utilitzar en centres amb diferents sistemes de puntuació.

5.1.4 Panell d'administració

Tot i que no s'ha desenvolupat un portal específic de *backof- fice* que permeti la gestió per part de l'administració dels centres, s'ha configurat el sistema d'administració que pro- veeix Django per a que es pugui utilitzar i gestionar de ma- nera ràpida i fàcil tota la informació del Smart Campus. A la figura 8 es pot veure un exemple de la pàgina principal de l'administrador de Django des del que es pot crear i mo- dificar tota la informació que hi ha desada. Per tant, serveix com a panell d'administració de Smart Campus.

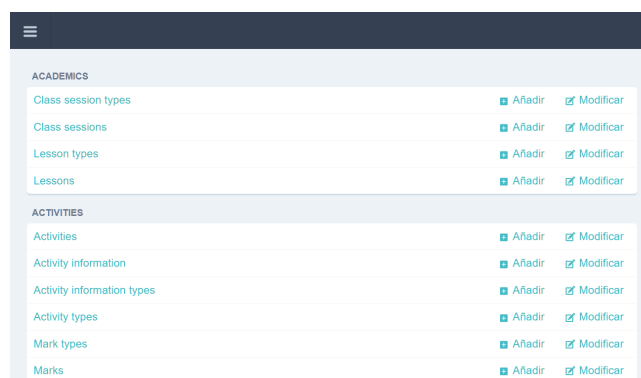


Fig. 8: Llista de models vista a l'administrador de Django

5.1.5 Tractament de dades

De cara a les parts IV, V i VI s'han sol·licitat dades anònimes a l'Oficina de Gestió de la Informació i de la Documentació (OGID) de la UAB. Per tal de poder con- nectar el nucli d'intel·ligència artificial amb el Smart Cam- pus, s'han adaptat els models de Django per poder guar- dar aquestes dades i s'ha fet un script que les carregui. Per tant, Smart Campus s'ha provat amb dades anonimitzades de més de 14.000 estudiants d'un total de 58 titulacions (grau i màster) de la Facultat de Ciències i l'Escola d'En- giñeria de la UAB.

Aquestes dades venien repartides en cinc fitxers d'Excel diferents que no es corresponien en cap cas amb un sol mo- del dels implementats a Django. Per tant, s'han estudiat les dades proporcionades per l'OGID i s'han incorporat als models en què s'ha considerat que encaixaven més. Un cop adaptats els models s'ha procedit a carregar les dades des dels fitxers.

El procés de lectura de dades, s'ha realitzat utilitzant la llibreria Pandas, que s'utilitza típicament per a treballar amb grans volums de dades. Un cop carregades les dades s'ha fet una preparació ràpida per adaptar-la als models de Django, ja que no totes coincidien amb els models que s'- havien definit per Smart Campus.

Un cop fets tots els scripts del codi, s'ha definit una co- manda de Django per poder realitzar aquesta lectura de ma- nera ràpida des del terminal i que les dades s'afegeixin al web.

Per poder tenir una instal·lació de Smart Campus comuna amb la que poder fer proves en un entorn local i, alhora, en el servidor, s'han desat totes les dades guardades en un format que es pugui importar i exportar.

5.2 Exposició de les dades

Com s'ha comentat prèviament, a Smart Campus hi ha un servidor de *backend* i un de *frontend* que es comuniquen a través d'una API de tipus GraphQL. El que s'ha realitzat ha sigut exposar totes les dades que hi ha al *backend* per a que puguin obtenir-se i mostrar-se des d'un altre servidor.

El fet d'utilitzar GraphQL ha permès desenvolupar una API dinàmica, molt més ràpida i que en un futur reduirà els costos de manteniment i facilitarà la seva integració.

La gestió de les dades es fa amb Graphene al *backend* i la lectura amb Apollo al *frontend*. L'exposició de les dades no només es fa per per connectar Django i React, també es fa per permetre que es sincronitzi Smart Campus amb Talentua. Per aconseguir-ho, s'ha definit l'API que exposa les dades que es requiriran a Talentua.

Es va intentar fer una implementació genèrica al *backend* per tal que pogués reutilitzar-se, però no és possible fer-ho amb Graphene pel seu funcionament. El que sí s'ha fet és definir el codi de manera que sigui fàcil d'adaptar a nous models que es vagin creant i es vulguin exposar a través de l'API de GraphQL.

S'ha fet un treball molt exhaustiu per assegurar la seguretat de les dades. Tots els detalls d'aquesta implementació es troben descrits a la part II pel Xavier Velasco.

5.3 Frontend

5.3.1 Disseny

A nivell del *frontend* s'ha realitzat una feina molt laboriosa de disseny que ha ocupat dos esprints. Inicialment, es van realitzar els dissenys utilitzant AdobeXD però, donat que React es basa en components, es va optar per canviar d'eina de disseny a Figma.

S'han realitzat *mockups* interactius amb Figma on s'han definit individualment els components que s'havien d'utilitzar. Treballar d'aquesta manera fa que, tot i que en un principi s'hagi de dedicar més temps a crear els components, després sigui més fàcil i ràpid crear les diferents pantalles i que aquestes siguin consistents entre si.

Fer un disseny tan elaborat ha fet que no es puguin desenvolupar algunes idees que han sorgit com, per exemple, un traductor de notes universal per tal de facilitar la lectura d'expedients d'estudiants d'Erasmus. Hi ha funcionalitats, com els esdeveniments repetitius o els recordatoris dels mateixos, que no s'han pogut implementar. Per tant, s'ha optat perquè el *frontend* no estigui completament desenvolupat i faltin algunes parts a canvi de tenir un disseny més sòlid, consistent i, sobretot, amb una experiència d'usuari més satisfactòria gràcies al fet d'haver invertit hores al disseny. A les figures 9 i 10 es pot veure l'exemple d'una pàgina en el *mockup* i la implementació amb React que s'ha realitzat.

5.3.2 Implementació

El *frontend* s'ha implementat utilitzant React i s'ha començat definint i creant els diferents components seguint el mateix procés que s'havia seguit a Figma.

S'han creat diverses pantalles i s'ha optat per prioritzar les que apropen Smart Campus a ser un MVP, com la pàgina principal, el perfil d'estudiant (figura 10), llistat de graus, assignatures, entregues, etcètera. Moltes d'aquestes

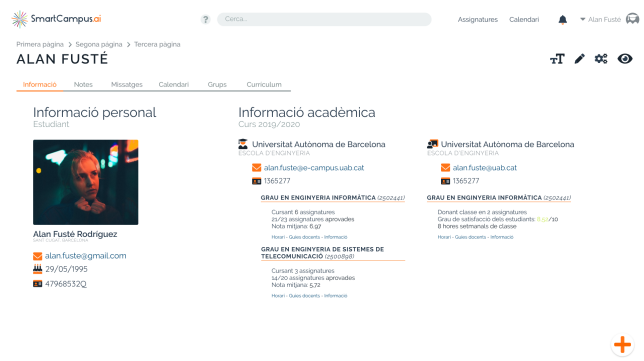


Fig. 9: Mockup de la pàgina d'usuari

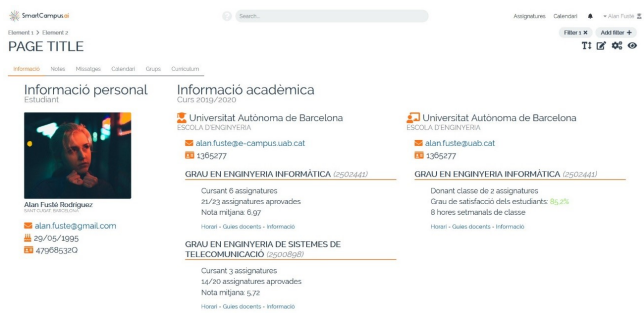


Fig. 10: Implementació de la pàgina d'usuari

pàgines hauran de completar-se a mesura que s'avanci en el desenvolupament i, en una fase posterior al MVP, analitzar-se per part d'un expert en UX/UI.

Com s'han incorporat les dades de l'OGID i s'ha realitzat una primera versió dels informes de les assignatures (figura 11), s'han creat pàgines per veure unes estadístiques bàsiques dels diferents graus amb el progrés de les notes per assignatura, així com llistats d'estudiants i de graus. Per tant, s'ha fet una primera aproximació al que seria un compte de direcció.



Fig. 11: Informe d'una assignatura

Amb aquestes funcionalitats, s'ha volgut mostrar la diversitat d'usuaris que poden fer servir Smart Campus: estudiants, professors i direcció. Per tant, s'ha optat per mostrar el potencial de la plataforma.

Per últim, com a element diferenciador, s'ha decidit incloure uns botons flotants a la zona inferior dreta que permeten fer diferents accions des de qualsevol lloc del web. Un estudiant, per exemple, pot realitzar una entrega no només des de la pàgina de la mateixa (el que suposa haver de fer un munt de clics), sinó que des de qualsevol lloc pot fer-ho i se li obre un *modal* (figura 12) on pot completar el procés fàcil i ràpidament. Les accions que es poden rea-

litzar utilitzant aquests botons són personalitzades per cada rol. En el cas d'un professor pot enviar un missatge als estudiants, crear una entrega o pujar noves notes. Es creu que aquest element tant simple serà, en un futur, un dels més importants del *frontend*, ja que permetrà agilitzar molt l'ús del LCMS.

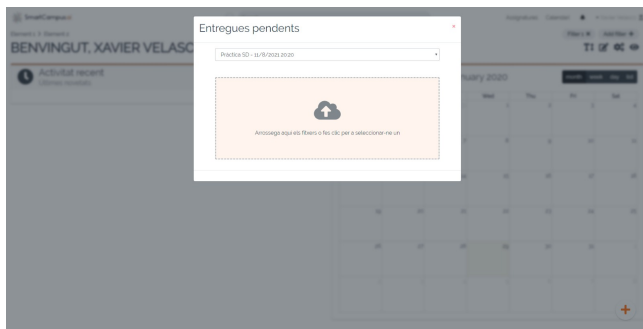


Fig. 12: Modal d'una entrega

6 CONCLUSIONS

L'objectiu global d'aquest treball era aconseguir un MVP de Smart Campus, una versió funcional que permetés fer proves en algun centre d'estudis petit. Això no s'ha aconseguit, però sí s'ha estat molt a prop d'aconseguir-ho i s'han avançat en aspectes que han permès tenir una base sòlida d'un LCMS que sembla que té recorregut per davant.

A la secció 2 s'han definit vuit funcionalitats requerides per a tenir un MVP atractiu i que es pogués provar en un entorn de producció petit. Hi ha dues funcionalitats que no s'han pogut abordar en cap aspecte: el sistema d'enquestes i el mòdul de denúncia anònima (enfocat a casos d'assetjament escolar). Tot i que no era primordial en un entorn universitari, es va prioritzar per tal de diferenciar Smart Campus d'altres LCMS com Moodle però, davant la falta de temps, aquestes dues funcionalitats s'han descartat i s'ha optat per avançar en la resta. D'altres elements s'han completat com a mínim al *backend* tal com s'havien plantejat.

Algunes funcionalitats com el calendari i els esdeveniments són funcionals però no estan acabades al detall que s'havia proposat. Tot i així, sí s'han establert les bases per a poder acabar-les durant els propers mesos. Cal recordar que aquests són només els primers esprints, per tant, els desenvolupaments seguiran en els propers mesos.

El repositori de material no s'ha inclòs a nivell de *frontend* però sí està completat a nivell de *backend* i amb un nivell més complet del previst al comptar amb elements com el control de versions.

La sincronització amb Talentua era una funcionalitat fonamental des del punt de vista de projecte, ja que era la base per enllaçar les tres primeres parts amb les següents, perquè es llegiran les dades directament de Smart Campus. Gràcies a l'ús de GraphQL s'ha pogut completar totalment aquesta funcionalitat. A més a més, el fet d'incorporar les dades de l'OGID ha permès assolir fites més enllà de les previstes, creant noves pantalles i incorporant una nova visió que no estava prevista pel marc del TFG: la direcció. D'aquesta manera s'han mostrat funcionalitats d'estudiants, professors i direcció/administració. El fet incloure dades reals i tan nombroses ha permès mostrar que la plataforma és

capaç de funcionar plenament tot i comptar amb milions de dades guardades.

Els objectius marcats inicialment van ser molt ambiciosos però, probablement, poc realistes. Els mòduls d'enquestes i denúncia anònima fan més atractiu un producte però no han de formar part d'un MVP, ja que no són estrictament necessaris.

En altres funcionalitats com les notes el fet de plantejar un mòdul tan complet com un full de càlcul *online*, que podria ser un producte per si mateix, va ser excessiu. De totes maneres el fet de marcar objectius ambiciosos ha fet que el ritme de treball sigui alt i exigent en tot moment, ja que no hi ha hagut "marge d'error".

Els informes de les assignatures és un tret fonamental de Smart Campus i sí s'ha pogut completar una primera versió bàsica, que deixa veure les línies futures d'aquesta funcionalitat. També cal dir que no es troba al nivell previst i queda molta feina per fer.

Tot i que no eren funcionalitats llistades en els objectius, hagués estat interessant comptar amb un sistema de notifiacions o permetre l'accés a fitxers d'altres graus dins d'una mateixa universitat. Aquest segon punt no s'ha abordat, ja que no s'ha pogut aprofundir en els rols i el sistema de permisos tant com caldria per tenir una funcionalitat així.

Per tant, tot i que no s'ha seguit al peu de la lletra el llistat de funcionalitats que es van definir als objectius inicials, sí s'han completat molts i s'han realitzat tasques que no estaven previstes però que s'ha considerat que aportaven més valor a un MVP. La definició dels models ha portat a tenir una arquitectura sòlida i robusta que es creu que permetrà que s'utilitzi Smart Campus en diferents entorns més enllà del que s'ha enfocat (UAB) durant aquesta primera versió. El temps dedicat al disseny del *frontend* va en la mateixa línia que la inversió de temps feta en el *backend* i s'ha buscat primar la qualitat per sobre de la quantitat, optant per fer menys pantalles però més ben treballades.

Com a balanç global es valora positivament aquests deu primers esprints, ja que es considera que s'ha fet molta feina i s'ha intentat que estigui el millor feta possible. El disseny ha endarrerit la planificació respecte a les pretensions que es tenien, però la implementació del *frontend* ha sigut ràpida i, tot i que hi ha carències en el disseny, es considera que és suficient per un MVP. Les claus d'aquests TFG passaven primer de tot per tenir un *backend* sòlid i robust, que fos més potent que el de Moodle i, en segon lloc, per donar una experiència d'usuari molt millor que la de Moodle. Tot i que queda molta feina per fer, el balanç és positiu i s'està demostrant que la tria del *stack* tecnològic ha sigut l'adequada.

Un factor que ha esdevingut importantíssim és la connexió entre plataformes i l'ús de GraphQL ha estat un gran encert. Això ha permès una gran comoditat i, a nivell de rendiment, es té previst obtenir uns resultats molt positius [6]. Aquest no era un factor clau per optar per GraphQL enlloc de l'API REST, ja se sabia que el podia millorar, tot i que no era el principal motiu, però ara es creu que serà un factor clau a nivell d'experiència d'usuari i que pot donar un gran avantatge respecte Moodle i d'altres LCMS. Alhora, tot i que no és un plantejament central de la solució, permet reduir dràsticament el consum energètic i, en conseqüència, l'empremta de carboni de Smart Campus.

Tot i que els objectius que es marcaven al començar eren

de funcionalitats i el marc del projecte és l'emprenedoria, cal destacar que s'ha fet una anàlisi de l'ètica de les solucions que s'han implementat. No es vol fer una plataforma només des d'una visió tecnològica i d'emprenedoria, sinó que sigui responsable i respecti els drets i llibertats dels usuaris. S'ha pres nota de solucions tecnològiques que han afectat a la privacitat dels usuaris i, per això, Talentua s'està construint tenint-ho en compte.

A nivell comercial, posant el focus en Smart Campus, es creu que hi ha un forat de mercat en els LCMS SaaS. És possible que amb una solució escalada globalment i amb ingressos provinents de vies diverses com Talentua i les llicències, però mai de la comercialització de dades, o amb una solució gestionada a nivell de sistema educatiu es pugui oferir un LCMS per un cost igual o inferior al que li suposa avui en dia Moodle a la UAB. Per una universitat pública el cost de les llicències de Blackboard és totalment privatiu i, per tant, tot i ser una tecnologia més potent que Moodle, no és viable utilitzar-la i, per tant, no està ocupant aquest forat de mercat.

7 FUTUR

En un futur, a mesura que augmenti el coneixement de l'equip vers React i GraphQL, es podran exprimir més aquestes tecnologies. La propera fita en aquest sentit serà treure suc a les noves característiques d'asincronia que incorpora Django 3 i integrar-les amb React 17 i GraphQL. També, en algun moment de 2020, actualitzar Bootstrap a la versió 5 que, com no comptarà amb jQuery, serà més ràpid i lleuger.

A nivell del Smart Campus i deixant de marge les parts IV, V i VI del treball, queden més de cinquanta tasques entre les que es van descartar i les que s'han hagut de deixar pendents. Hi ha tasques de funcionalitats com un xat, un sistema de Q&A i d'altres de molt volum com la realització d'un *dashboard* per la direcció dels centres o un *backoffice* d'administració. També queden tasques que pertanyen a fases més avançades com integració amb Moodle o Dropbox.

A la secció 2, en l'exposició dels objectius, es fa referència a la voluntat de fer un MVP per l'estiu de 2020 que serveixi a mode de beta per provar la plataforma de Smart Campus en un entorn controlat però de producció. El plantejament inicial és que Talentua esdevingui una societat limitada amb dues plataformes que es comercialitzin com a serveis.

En el món de l'emprenedoria hi ha un element que és comú en qualsevol projecte se segueixi o no *Lean Startup*⁴: utilitzar un *Business Model Canvas*[7] o, simplement, *canvas*. El *canvas* és una plantilla de gestió estratègica que s'utilitza per desenvolupar nous models de negoci. Aquest gràfic visual conté diferents seccions que descriuen, principalment, les propostes de valor, la infraestructura, els clients i les finances. El fet de ser visual i resumit el fa molt útil ja que ajuda a les empreses a tenir una visió ràpida i clara del seu model de negoci.

A l'apèndix A.2 es pot trobar el *canvas* de Talentua i Smart Campus, el qual s'ha anat actualitzant de manera constant durant prop d'un any i mig de manera regular i, per tant, és fruit de mesos de treball i reflexió.

⁴*Lean Startup* [8] és una metodologia de desenvolupament de negocis i productes. Consisteix en fer llançaments del producte de manera iterativa a mesura que es millora el producte en base a l'opinió dels clients.

8 AGRAÏMENTS

Al Xavi i el Dyhagho, per acceptar formar part d'aquest projecte quan us vaig reunir el juny de 2018 i per la vostra dedicació. Sé que de vegades sembla difícil i que potser no compensa, però valdrà la pena!

Al Rubén i, també, al Jose, al Carlos i al Dani, per totes les hores que heu dedicat del vostre temps en ajudar-nos, pel vostre acompanyament en aquest procés, pels consells, pels ànims i per donar suport a aquest projecte. Gràcies també, per demostrar que es poden iniciar projectes entre grups d'estudiants a una universitat pública com la UAB.

A la Neus, pel teu suport incondicional, per animar-me i recolzar-me amb aquest projecte des d'abans que comencés.

Al David, per les teves idees i el teu *feedback* sense el qual, de ben segur, Talentua no estaria concebut així.

Als meus pares, pel vostre suport i ànims per fer aquesta aposta i pel referent que heu sigut i sou.

REFERÈNCIES

- [1] Universitat Autònoma de Barcelona. "Pressupost 2019", Universitat Autònoma de Barcelona, Cerdanyola del Vallès, 2019.
- [2] C. Pino. "Costos de Blackboard Open LMS: Precios Actualizados al 2020". Internet: <https://www.comparasoftware.com/blackboard-open-lms/> [02 Feb. 2020].
- [3] Marketsandmarkets.com. "LMS Market Size by Solutions & Services - 2023". Internet: <https://www.marketsandmarkets.com/Market-Reports/learning-management-systems-market-1266.html> [2 Feb. 2020].
- [4] Researchandmarkets.com. "Learning Management System (LMS) Market to 2025 Global Analysis and Forecasts". Internet: https://www.researchandmarkets.com/research/3g2dlp/18_4_billion [2 Feb. 2020].
- [5] eLearning Industry. "Top 10 e-Learning Statistics for 2014 You Need To Know". Internet: <https://elearningindustry.com/top-10-e-learning-statistics-for-2014-you-need-to-know> [2 Feb. 2020].
- [6] Netflix. "Our learnings from adopting GraphQL". Internet: <https://medium.com/netflix-techblog/our-learnings-from-adopting-graphql-f099de39ae5f> [11 Des. 2019].
- [7] Villanueva, J. (2019). "Introducción al emprendimiento: Conceptos y metodologías para crear startups". 1ra ed. Amazon.
- [8] Ries, E. and San Julián, J. (2013). "El método Lean Startup: Cómo crear empresas de éxito utilizando la innovación continua". Deusto.

APÈNDIX

A.1 Anàlisi de tecnologies

A.1.1 PHP

És el llenguatge clàssic de *backend*. És el més utilitzat i el que té una comunitat més extensa. És senzill d'utilitzar i ens aporta tot el necessari per a abordar el projecte de forma ràpida i senzilla.

| Pros | Contres |
|-----------------------------|-----------------------------------|
| Velocitat de programació | Rendiment final |
| Senzill però potent | Canvis constants en el llenguatge |
| Comunitat extensa | |
| Integració perfecta amb SQL | |

TAULA 3: PROS I CONTRES DE PHP

Laravel és, actualment, el *framework* més utilitzat i amb la comunitat més gran i activa.

| Pros | Contres |
|--------------------------------|--|
| <i>Framework</i> més utilitzat | No l'utilitza cap companyia líder |
| Rendiment final acceptable | Canvis constants en el llenguatge |
| Velocitat de programació | Presenta limitacions en projectes grans |
| Orientat a MVC | Falta de suport complet a <i>PHPUnit</i> |
| Aprenentatge amigable | |

TAULA 4: PROS I CONTRES DE LARAVEL

Symfony és un *framework* format per mòduls separats que junts, funcionen perfectament. Té com a objectiu aconseguir la màxima estabilitat en projectes molt grans. Utilitzat per serveis com Spotify, BlaBlaCar, Pornhub, Trivago o Dailymotion.

| Pros | Contres |
|---------------------------------|------------------------|
| Molt escalable | Relativament complicat |
| Millor rendiment que la majoria | |
| Comunitat gran | |

TAULA 5: PROS I CONTRES DE SYMFONY

Phalcon és un *framework* basat en C que té com a punt destacable el poc consum de recursos i el bon rendiment que ofereix. S'instal·la com a extensió directa de Php.

A.1.2 JavaScript

JavaScript ofereix moltíssimes eines de tot tipus. Permet construir una aplicació web utilitzant un sol llenguatge tant per al *frontend* com per al *backend*, facilitant la programació. Ofereix un rendiment extraordinari en la majoria de casos, per sobre de la resta de llenguatges (excepte Go i

| Pros | Contres |
|--------------------------|-------------------------------------|
| Màxim rendiment | Debugging difícil |
| Consum mínim de recursos | Necessitat d'accés root al servidor |
| | Comunitat petita |
| | Relativament complicat |

TAULA 6: PROS I CONTRES DE PHALCON

C++). La gran majoria de webs modernes l'utilitzen, ja sigui en forma de llibreria *frontend* com React o en forma de *framework backend* com ExpressJS.

| Pros | Contres |
|--|-------------------------|
| Rendiment final | <i>Single-threaded</i> |
| Velocitat de programació | Configuració "complexa" |
| Permet escriure l'aplicació sencera en un sol llenguatge | Canvis continus |
| Modern | Poc madur |
| Comunitat més activa | |

TAULA 7: PROS I CONTRES DE JAVASCRIPT

React és una llibreria de JavaScript per *frontend* que permet crear la interfície d'usuari a partir de components reutilitzables. Ofereix un control molt gran sobre l'estat de l'aplicació. Utilitza la sintaxi JSX que després és transformada a ES6. Netflix, Yahoo, Dropbox, Airbnb... són un exemple de companyies que utilitzen React.

| Pros | Contres |
|-----------------------------------|---|
| Llenguatge base senzill | Necessitat de llibreries externes per a projectes grans |
| Velocitat de programació | Rendiment final |
| Control de l'estat de l'aplicació | Configuració extensa |
| React Native | |

TAULA 8: PROS I CONTRES DE REACT

Angular és, segurament, el *framework* de *frontend* més complet de JavaScript. Està enfocat a oferir el màxim rendiment en grans projectes. Semblant a VueJS però amb certs extres que faciliten el desenvolupament i milloren el rendiment. Google, Paypal i Nike són algunes de les empreses que l'utilitzen.

| Pros | Contres |
|---------------------------|---|
| Enfocat a grans projectes | Més complex que la resta de <i>frameworks</i> |
| Rendiment final | |
| <i>Direct DOM</i> | |
| <i>Fenomenal CLI</i> | |
| Angular Ionic | |

TAULA 9: PROS I CONTRES D'ANGULAR

VueJS (*frontend*) va ser el *framework* millor valorat a GitHub l'any 2017. Aporta les tècniques bàsiques per a implementar aplicacions web de forma ràpida i senzilla. S'u-

tilitza principalment a Àsia i l'utilitzen empreses com Alibaba, Xiaomi o Adobe.

| Pros | Contres |
|----------------------------|---|
| Senzill | Necessitat de llibreries externes per a projectes grans |
| Velocitat de programació | Comunitat petita |
| Vue Framework ⁷ | |

TAULA 10: PROS I CONTRES DE VUEJS

ExpressJS és un *framework* de *backend* que ofereix molt bon rendiment al ser minimalista però que alhora deixa molta flexibilitat per afegir els mòduls necessaris per al projecte en qüestió. Es pot complementar amb qualsevol dels altres *frameworks frontend* ja mencionats. És, segurament, el *framework backend* més utilitzat després de Ruby on Rails.

| Pros | Contres |
|------------------------------------|--|
| Enorme comunitat a grans projectes | S'ha d'entendre bé la naturalesa <i>event-driven</i> |
| Rendiment final | |
| Flexible | |
| Velocitat de programació | |

TAULA 11: PROS I CONTRES DE EXPRESSJS

A.1.3 Java

Java és el llenguatge més utilitzat en tot tipus de programes. No és el llenguatge més modern, però segueix tenint un bon rendiment si es programa bé i, a més a més, té una de les comunitats més grans.

| Pros | Contres |
|-------------------------------------|--|
| Rendiment final (si es programa bé) | Lent de programar |
| Comunitat enorme | <i>Overkill</i> en la majoria de casos |
| Robust | Complex |
| Fàcil de debugar | |
| Enfocat a projectes grans | |

TAULA 12: PROS I CONTRES DE JAVA

A.1.4 Python

Python és un llenguatge molt enfocat a projectes que requereixen un *backend* molt complex. Es pot utilitzar juntament amb altres llenguatges *frontend* com JavaScript + Angular, per posar un exemple. S'utilitza com a substitut del PHP. És utilitzat per la majoria de grans empreses.

Django és un *framework* de *backend* de Python que permet crear REST APIs amb *backend* complexos de forma ràpida i senzilla i, a més a més, compta amb un sistema de *templates* que permet la creació de *frontend*. Té una molt bona comunitat i molts recursos dels quals aprendre coses noves. Instagram, Coursera i BitBucket utilitzen Django en les seves aplicacions web.

Pyramid és un *framework* lleuger molt semblant a Django però més senzill i flexible. Normalment s'amplia amb

| Pros | Contres |
|---|--|
| Velocitat de programació | Necessitat d'un servidor que l'accepti |
| Senzill | |
| Comunitat enorme | |
| Permet crear <i>backends</i> complexos de forma fàcil | |

TAULA 13: PROS I CONTRES DE PYTHON

| Pros | Contres |
|--|------------------------------|
| Creació d'interfície d'administrador molt senzilla | Pitjor rendiment que Pyramid |
| <i>Templates</i> | |
| Comunitat força gran | |

TAULA 14: PROS I CONTRES DE DJANGO

llibreries externes en projectes grans per poder utilitzar *templates* o tecnologies per l'estil. Mozilla és una de les empreses que utilitza aquest *framework*.

| Pros | Contres |
|-----------------------------|---|
| Millor rendiment que Django | Necessitat de llibreries externes per a projectes grans |
| Flexible | Comunitat petita |

TAULA 15: PROS I CONTRES DE PYRAMID

A.1.5 Ruby

Ruby ve a ser l'alternativa moderna a PHP. És un llenguatge complex però que ofereix molta potència un cop el domines. Té un rendiment molt similar a Python, però disminueix força en projectes grans si no està perfectament programat. És un dels més utilitzats, sobretot per programadors PHP que veuen que s'estan quedant enrere. Twitter utilitzava Ruby on Rails com a *framework backend* fins fa pocs anys.

| Pros | Contres |
|--|------------------------------------|
| Velocitat de programació (un cop has assimilat bé el llenguatge) | Menys escalabilitat que la majoria |
| <i>Multi-threaded</i> | Més aviat complex |
| Codi net, flexible, compacte i fàcil de mantenir | Rendiment final |

TAULA 16: PROS I CONTRES DE RUBY

Ruby on Rails és el *framework* més utilitzat, amb diferència, de Ruby. És molt complet i facilita molt la feina a l'hora de programar. Té una comunitat enorme al ser tan utilitzat. Webs com GitHub, Twitch, Airbnb o Twitter l'han fet servir.

| Pros | Contres |
|--------------------------|-----------------------|
| Velocitat de programació | Rendiment final |
| Comunitat enorme | Falta de flexibilitat |

TAULA 17: PROS I CONTRES DE RUBY ON RAILS

A.1.6 Go

Go és l'alternativa de Google al llenguatge C. És un llenguatge força senzill que té moltes de les característiques de C però que n'implementa algunes com el *garbage collector* que simplifiquen molt la programació. És un llenguatge compilat així que ofereix un rendiment molt bo. Moltes empreses com Twitter, Facebook o Dropbox estan començant a utilitzar-lo.

| Pros | Contres |
|--|------------------|
| Lleuger i senzill | Poc flexible |
| Rendiment final | Comunitat petita |
| Modern | |
| Molta escalabilitat | |
| Molt madur pel poc temps que fa que existeix | |

TAULA 18: PROS I CONTRES DE GO

A.1.7 Scala

Llenguatge dissenyat per a ser una versió millorada de Java. Ofereix un molt bon rendiment i una escalabilitat fenomenal a canvi d'una programació una mica més lenta i complicada que en la resta de llenguatges. És un llenguatge modern utilitzat per Twitter, Tumblr i Soundcloud, per exemple.

| Pros | Contres |
|--------------------------|---|
| Més senzill que Java | Orientació híbrida (funcional i objectes) |
| Rendiment final | Comunitat petita |
| Escalabilitat | Difícil d'aprendre |
| Velocitat de programació | Versions noves incompatibles amb les antigues |

TAULA 19: PROS I CONTRES DE GO

A.1.8 Frameworks de CSS

Bootstrap és el *framework* més utilitzat al món amb prou diferència. És el que disposa d'una comunitat més gran i d'una millor documentació. És senzill d'aprendre i permet programar a una molt bona velocitat. És una mica més pesat que la resta ja que utilitza JS a més de CSS però ofereix moltíssimes possibilitats. Depèn de llibreries com jQuery o Popper, d'aquesta manera és no només un *framework* de CSS, sinó un *framework* de *frontend* en general.

Bulma és el *framework* de moda. La seva comunitat ja és de les més grans i no para de créixer. És un *framework* que treballa 100% amb CSS, per tant és molt lleuger i ofereix un rendiment excepcional. Ofereix moltes possibilitats i també l'opció de modificar més de 200 variables Sass per a personalitzar el *framework* com sigui necessari. És molt simple d'utilitzar i aprendre, i treballar amb ell és molt fàcil i còmode. Està basat en Flexbox i pensat per a mòbil, així que és totalment *responsive* sense haver d'incloure tant codi com a Bootstrap o d'altres.

Foundation és un dels *frameworks* més utilitzats per les grans empreses. És *mobile-first* i molt personalitzable. Fa

ús de JS junt amb el CSS per oferir moltes més possibilitats sacrificant una mica de rendiment. Depèn de tecnologies com modernizr, HTML5 Boilerplate o Normalize per a funcionar. Té una documentació excel·lent i una comunitat molt gran i activa. Disposa de variables Sass per a personalitzar. És prou ràpid de programar i

Materialize és el *framework* basat en Material Design més utilitzat actualment. Barreja CSS amb JS per oferir uns resultats immillorables i una velocitat de producció prou bona. Té una de les comunitats més grans i disposa d'una molt bona documentació que permet treballar ràpid i sense problemes.

Material components és el *framework* creat pels enginyers de Google per a implementar Material Design de forma modular. Material Components s'apropa més a ser un *framework* de *frontend* de JS que no pas a un *framework* de CSS, però l'objectiu segueix sent donar estil a la pàgina de forma ràpida i senzilla. S'incorpora perfectament amb la majoria de *frameworks* de JS com Angular o React, cosa que permet treballar sense problemes i molt còmodament. És més complicat que la resta però la boníssima documentació ho pot arribar a compensar. La seva comunitat no és la més gran però hi ha prou gent que l'utilitza.

Semantic és el *framework* que lluita pel relleu de Bootstrap com a número 1. És més modern, ofereix més possibilitats, el seu CSS és més senzill i fàcil d'entendre i el seu rendiment és millor, però és més difícil de configurar i requereix més coneixements de JS que Bootstrap. Ofereix moltíssimes possibilitats, un bon rendiment i molts temes ja configurats. També té un munt de variables CSS per a personalitzar els temes. Cada cop més usuaris de Bootstrap estan començant a utilitzar Semantic, així que la comunitat no para de créixer i disposa també d'una documentació detallada que ajuda molt a l'hora de programar.

A.1.9 Bases de dades

MySQL és una base de dades pertanyent a Oracle fa pocs anys. És la més utilitzada juntament amb la pròpia base de dades de Oracle. Porta molts anys al mercat, és simple, funciona bé i té una comunitat enorme al darrere. Ve configurada per defecte en molts paquets com LAMP. Disposa de moltíssimes eines de tercers i és compatible amb qualsevol servidor. Va ser creada per a donar un rendiment màxim en projectes que de mida mitjà-petit, però amb les últimes versions ofereix un rendiment excel·lent en qualsevol entorn, consumint molts pocs recursos.

MariaDB està feta pels creadors de MySQL, va aparèixer com una millora de MySQL quan aquesta última va ser comprada per Oracle. És molt semblant a MySQL, igual de senzilla però més moderna, amb millor rendiment i més utilitats. Porta poc temps al mercat però té una comunitat molt gran ja que l'utilitzen les empreses més importants.

PostgreSQL està enfocada a projectes grans, és una de les bases de dades més antigues i més utilitzades en els seus millors dies. No té tantes característiques modernes com MariaDB ni és tan simple i fàcil d'utilitzar com MySQL, però és molt robusta i dona un molt bon rendiment que es diferencia de les altres quan la base de dades augmenta molt el mida. És menys utilitzada, té menys comunitat i menys eines de tercers, però és més escalable.

A.2 Bussiness Model Canvas

| Socis clau | Activitats clau | Proposta de valor | Relació amb el client | Segment de clients |
|---|--|--|---|---|
| <ul style="list-style-type: none"> Governos autonòmics que busquin millorar l'educació pública i potenciar els nous talents Empreses arrelades en el sector educatiu que no disposin de plataformes digitals i vulguin expandir-se o millorar les seves plataformes (per exemple, JobTeaser, Jobandtalent o SigmaAIE) Entitats certificadores per les que la seva marca és essencial Empreses del sector tecnològic que vulguin expandir-se al sector de la intel·ligència artificial, l'educació o la selecció de personal | <ul style="list-style-type: none"> Associar-se amb les universitats Arribar a acords amb les institucions governamentals Realitzar acords amb grans empreses de recursos humans Desenvolupar intel·ligència artificial acurada i viable per entorns de producció Crear dues plataformes web modernes i escalables | <ul style="list-style-type: none"> Es realitza el millor <i>matching</i> possible entre estudiants i empreses Ús i recollida de dades poc comunes i de qualitat Milliores per a centres educatius: <ul style="list-style-type: none"> Estudiants de més qualitat Menor cost de la plataforma digital Més facilitats per als docents Noves i suculentos vies d'ingrés Reducció dels costos de reclutament de les empreses (directa i indirectament) Talentua se centra en el treball qualificat i en la promoció del talent real (aquell certificat) Ús de tecnologia escalable i moderna en el desenvolupament de les plataformes web i de la intel·ligència artificial Ambdues plataformes són ampliables i es poden desenvolupar mòduls a mida Talentua dona accés al mercat empresarial i això la situació en una posició clau per a realitzar consultoria Talentua i Smart Campus són respectuoses amb totes les dades recollides i no es cediran a tercers | <ul style="list-style-type: none"> A través d'una plataforma on podran indicar quin perfil busquen (Talentua) Amb un equip de relacions públiques i gestió de contractes amb les institucions educatives (Smart Campus) La relació amb el client és fonamental al tractar-se d'un servei de qualitat i parcialment personalitzat | <ul style="list-style-type: none"> Empreses que busquen personal qualificat, els <i>early adopters</i> són empreses de recursos humans (Talentua) Centres educatius tant públics com privats (Smart Campus) Entitats certificadores de qualsevol tipus (per exemple, escoles d'idiomes, cursos online, acadèmies, etc.) que volen potenciar el reconeixement dels seus estudiants i donar valor a nivell laboral per la seva formació (Talentua) Empreses que busquin plataformes per a realitzar cursos online interns (Smart Campus) Els <i>early adopters</i> són cursos de FP, Masters i centres d'estudis privats (Smart campus). Empreses o centres que vulguin serveis personalitzats i ampliacions del programari (Smart Campus, Talentua, consultoria) Usuaris: <ul style="list-style-type: none"> A Smart Campus hi ha: estudiants, personal administrador, docents i direcció dels centres d'estudis. A Talentua hi ha: personal de les empreses, perfils institucionals i estudiants. |
| Estructura de costos | | Fonts d'ingressos | | |
| <ul style="list-style-type: none"> Personal tècnic Personal de gestió Personal de recursos humans Personal de ventes Infraestructura de servidors (servei de núvol extern) Màrqueting digital per Talentua Domínis Llicències de programari Marketplace d'aplicacions (AppStore, etcètera) | | <ul style="list-style-type: none"> Empreses o institucions: <ul style="list-style-type: none"> Per accedir a Talentua (model <i>Freemium</i>) Centres educatius, acadèmies o governs: <ul style="list-style-type: none"> Per disposar del Smart Campus (model <i>Freemium</i>) Per desenvolupament de noves funcionalitats de Smart Campus Les acadèmies d'oposicions poden certificar als estudiants que es preparen. Per tant, tot i que no obtinguin la plaça a la que opositen haver estudiat els servirà perquè ponderarà a Talentua. Això suposa un valr afegit tant per l'acadèmia com per l'estudiant Per obtenir informació de quins aspectes s'haurien de potenciar de cara a enfocar els estudis superiors i els plans d'estudis dels mateixos Entitats certificadores <ul style="list-style-type: none"> Per introduir títols a Talentua (model <i>Freemium</i>) Consultoria <ul style="list-style-type: none"> Per realitzar projectes de consultoria (principalment en intel·ligència artificial) en el marc de crear noves solucions d'utilitat per les empreses i d'altres basades en les nostres plataformes | | |

Fig. 13: Bussiness Model Canvas de Talentua